

A work flow orders data flows and the operations that support them. It also defines the interdependencies between data flows. For example, if one target table depends on values from other tables, you can use the work flow to specify the order in which you want Data Services to populate the tables.

You can also use work flows to define strategies for handling errors that occur during project execution, or to define conditions for running sections of a project. A data flow defines the basic task that Data Services accomplishes, which involves moving data from one or more sources to one or more target tables or files.

You define data flows by identifying the sources from which to extract data, the transformations the data should undergo, and targets.

Defining Projects and Jobs

A project is the highest-level object in Designer. Projects provide a way to organize the other objects you create in Designer.

A job is the smallest unit of work that you can schedule independently for execution. A project is a single-use object that allows you to group jobs.

For example, you can use a project to group jobs that have schedules that depend on one another or that you want to monitor together.

Projects have these characteristics:

- Projects are listed in the Local Object Library.
- Only one project can be open at a time.
- Projects cannot be shared among multiple users.

The objects in a project appear hierarchically in the project area. If a plus sign (+) appears next to an object, you can expand it to view the lower-level objects contained in the object. Data Services displays the contents as both names and icons in the project area hierarchy and in the workspace. Jobs must be associated with a project before they can be executed in the project area of Designer.



Note:

By default the Designer displays the first 17 characters of an object name. This can be changed in Options.

Using Work Flows

Work flows are not required except in a few circumstances. Jobs with data flows can be developed without using work flows. However, one should consider nesting data flows inside of work flows by default. This practice can provide various benefits. Consistent use of work flows make jobs more adaptable to additional development and/or specification changes.

For instance, if a job initially consists of four data flows that are to run sequentially, they could be set up without work flows. But what if specification changes require that they be merged into another job instead? The developer would have to replicate their sequence correctly in the other job. If these had been initially added to a work flow, the developer could then have simply copied that work flow into the correct position within the new job. There would be no need to learn, copy, and verify the previous sequence. The change can be made more quickly with greater accuracy.